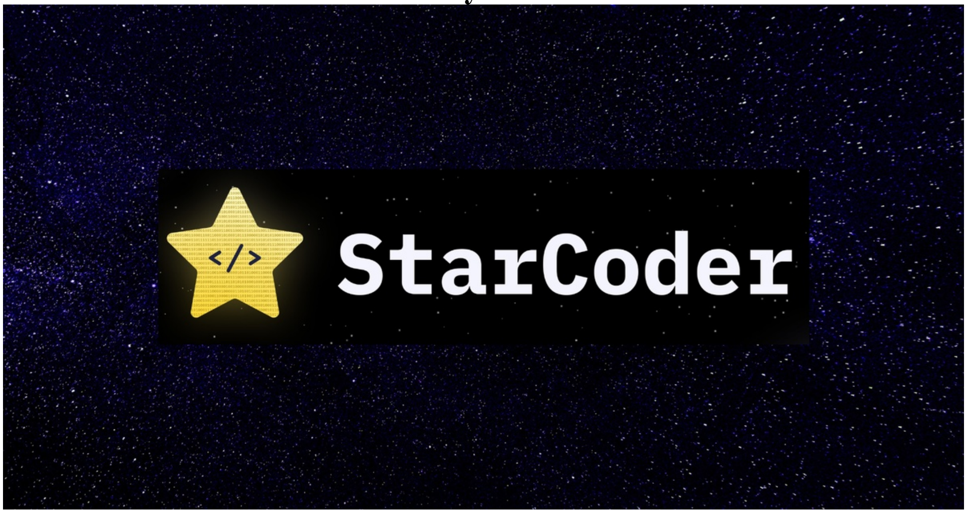
Summary of Results



I used the **TinyStarCoder** model on a dataset made from my own code project for machine learning and AI I created for recognizing stars and galaxies. Since TinyStarCoder is a small model, I wasn't expecting perfect results, but it actually did alright for a simpler model. It didn't always hit the exact code I wanted, often missing some words or adding too much, but it performed pretty reasonably overall.

Here's a quick overview of what I managed to do:

- Made a dataset from my own code, with each example split into prefix (code before), middle (missing code), and suffix (code after).
- Ran TinyStarCoder on these examples to fill in the missing middle part.
- Compared the model's completions with the actual code using some automatic metrics like BLEU, chrF, ROUGE, and exact match.
- Also reviewed some completions by hand and saved the incorrect ones for more analysis.

In the end, TinyStarCoder showed some good signs for such a small model. The results show it could probably get better if it had more context or fine-tuning, but overall, it handled this test reasonably well. I saved the incorrect completions separately to study what parts the model got wrong, which should help guide more improvements.

After running the analysis on the model's code completions, here are the key findings. Overall, the model didn't perform too well, and it seems there's a lot of room for improvement. I think we could use more robust model because I used tiny_starcoder that might not perform too well. Here's a quick breakdown of each metric I used and what the results showed.

Exact Match Rate: 8%
This means only 8% of the model's outputs exactly matched the expected code completion. Basically, it only got it perfectly right 8 times out of 100, which is not that bad as this is for exact matches.

Average BLEU Score: 0.0233
BLEU measures how similar the output is to the target by looking at matching sequences of words. A score this low shows that the model's completions often don't have matching word patterns with what was expected.

Average chrF Score: 23.89
This score tells us how similar the completions are on a character-by-character basis, not just words. A score around 23 (out of 100) is quite low, which means there’s only minor character overlap between the generated text and the target.

Average ROUGE-L Score: 0.2148
ROUGE-L measures the longest matching parts between the model’s completion and the target.

Average Similarity Score: 0.2403
This score is from comparing overall similarity between the model’s output and the target. A similarity score of 0.24 (out of 1) means the model's completions are 0.24 close to expected, structure-wise or word-wise.

Average Length Difference: 13.55 characters
On average, the model’s output is about 13-14 characters different in length compared to the expected output. This can mean that the model either adds too much or too little code.

Conclusion

We used the **TinyStarCoder** model for this task, which is a lightweight model. Given its small size, its expected that its performance wouldn’t match that of larger, more powerful models. However, considering the simplicity of TinyStarCoder, the results are actually quite good. The detailed incorrect completions are saved in a file, so looking through those might offer ideas on where the model went off track and help guide further improvements.